Project Report

XING XING *(z5142063)*

I. Introduction and Background

The main aim of this project is to classify of a images if there is a presence of pedestrians. This project is typically an image-processing and recognition project. According to the requirements of the assignment, is forbidden for using deep learning techniques, traditional feature extraction techniques (mainly KAZE and SIFT) are used. The KAZE algorithm was proposed by a French scholar in 2012 ECCV conference and is a more stable feature detection algorithm than SIFT. KAZE is named in honor of the pioneer of scale-space analysis, the Japanese scholar Lijima. KAZE is a homonym of 'wind' in Japanese. The implication is that just as the formation of wind is a non-linear flow of air in space, KAZE feature detection is a process of non-linear diffusion processing in the image domain. And it could save more image feature.

II. Method

I use google.colab platform because its easier to train with very big data, and it allows me develop the project incrementally.

```
[ ] import os
    from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

```
[ ] path = "/content/drive/My Drive"
    os.listdir(path)
```

```
['Getting started.pdf',
 'Colab Notebooks',
 'Individual_Component',
 'content',
 'Individual_Component.zip']
```

```
[ ] #os.makedirs('/content/drive/Individual_Component')
    print(os.path.exists('/content/drive/My Drive/Individual_Component/test'))
```

```
True
```

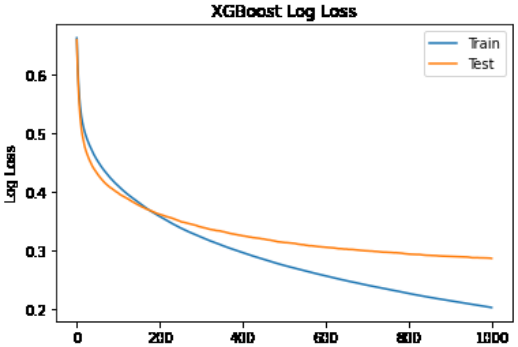
I choose the KAZE algorithm for the project because it does not cause boundary blurring and loss of details and is stable for training. In the case of information loss because of image blur, noise, and compression reconstruction, the robustness of KAZE features is much better than the others features or algorithms. However the KAZE is inferior to SIFT in scale invariance. And it need a little more detection time than SURF, but similar to SIFT. The KAZE algorithm mainly includes the following 4 steps: Construction of nonlinear scale space; Detection and precise positioning of feature points; Determination of the main direction of the feature point; Generation of feature descriptors.

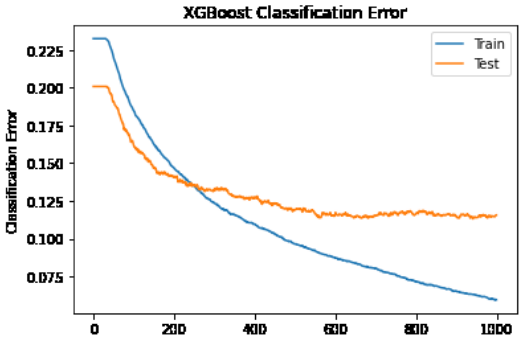
III. Experiment

Since the data is on the google drive already, I directly use the google colab platform as the computing resource for the project. I mainly use the open-cv calculation package. In order to compare the performance of the data on SIFT, I uninstalled the original open-cv version on the platform and installed open-cv-python == 3.4.2.16, open-cv-contrib-python == 3.4.2.16. The overall data processing pipeline is: Read and analyze the score. Convert pictures into KAZE features. Map the features to labels. Use XGboost to train the model, adjust the parameters of the model, and test on the test set. When reading and analyzing the data, I found that the number of positive samples in the training set is 7 + 8 + 8 = 23 and the number of negative samples is 50, which is a slight imbalance of the samples, but it is not enough to affect the model training which is ok, so When the machine memory allows (google colab can provide a large memory of 12G), I choose to read all the data instead of down sampling the data. After importing python.os and using the relevant tools of it, I can already read the files in batches and then use open-cv to calculate the KAZE characteristics of the files. In the process of calculating features, the core code is to find the key points of the image. The number of key points is generally related to the size of the image and the content of the image itself. All images in the project are the same size, so there is no need to resize and normalize the image. After obtaining the key points, sort the returned key points. The larger the value, the more important the key points are, so they should be ranked first. After finishing sorting, get the most important and first 32 key points. After obtaining 32 key points, the calculation of the descriptor vector is started. Expand the descriptor vector. If there are less than 32 descriptors, zeros are added after the feature vector. Now it have converted each picture into a 32*64 long vector, and corresponding the vector and the label to form a wide table. Use the wide table and XGboost for the next calculation. Because it is a binary classification problem, the loss function is binary: logistic. During the training process, the auc index is observed in real time. After the training is completed, the training and test sets are printed which is coming in a percentage.

IV. Results and Discussion

During the training of the XGboost model, I actually printed log loss and error, and set early stopping for preventing overfitting for each cases where the log loss did not improve for 50 rounds.





After tuning, we showed two tables, XGBoost Log Loss and XGBoost Classification Error, to show the finally training curve.

It can be found from the curve that the performance of the test set did not improve significantly after 500 rounds.

The final accuracy rate is 90%(between 88% and 92%) which seems to be an ok result.

```
[0]     validation_0-error:0.232458     validation_0-logloss:0.662967   validation_1-error:0.
Multiple eval metrics have been passed: 'validation_1-logloss' will be used for early stoppin

Will train until validation_1-logloss hasn't improved in 50 rounds.
[1]     validation_0-error:0.232458     validation_0-logloss:0.638376   validation_1-error:0.
[2]     validation_0-error:0.232458     validation_0-logloss:0.618162   validation_1-error:0.
[3]     validation_0-error:0.232458     validation_0-logloss:0.601158   validation_1-error:0.
[4]     validation_0-error:0.232458     validation_0-logloss:0.586832   validation_1-error:0.
[5]     validation_0-error:0.232458     validation_0-logloss:0.574487   validation_1-error:0.
[6]     validation_0-error:0.232458     validation_0-logloss:0.56383    validation_1-error:0.
[7]     validation_0-error:0.232458     validation_0-logloss:0.55492    validation_1-error:0.
[8]     validation_0-error:0.232458     validation_0-logloss:0.5473     validation_1-error:0.
[9]     validation_0-error:0.232458     validation_0-logloss:0.540646   validation_1-error:0.
[10]    validation_0-error:0.232458     validation_0-logloss:0.534635   validation_1-error:0.
[11]    validation_0-error:0.232458     validation_0-logloss:0.529513   validation_1-error:0.
[12]    validation_0-error:0.232458     validation_0-logloss:0.524863   validation_1-error:0.
.....

[989]   validation_0-error:0.060113     validation_0-logloss:0.203757   validation_1-error:0.
[990]   validation_0-error:0.060281     validation_0-logloss:0.203615   validation_1-error:0.
[991]   validation_0-error:0.059944     validation_0-logloss:0.203488   validation_1-error:0.
[992]   validation_0-error:0.059818     validation_0-logloss:0.203359   validation_1-error:0.
[993]   validation_0-error:0.059944     validation_0-logloss:0.203257   validation_1-error:0.
[994]   validation_0-error:0.05986      validation_0-logloss:0.203135   validation_1-error:0.
[995]   validation_0-error:0.059482     validation_0-logloss:0.203029   validation_1-error:0.
[996]   validation_0-error:0.05986      validation_0-logloss:0.20289    validation_1-error:0.
[997]   validation_0-error:0.059608     validation_0-logloss:0.202757   validation_1-error:0.
[998]   validation_0-error:0.059524     validation_0-logloss:0.202631   validation_1-error:0.
[999]   validation_0-error:0.059608     validation_0-logloss:0.202486   validation_1-error:0.
Accuracy: 88.43%
```

V.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989